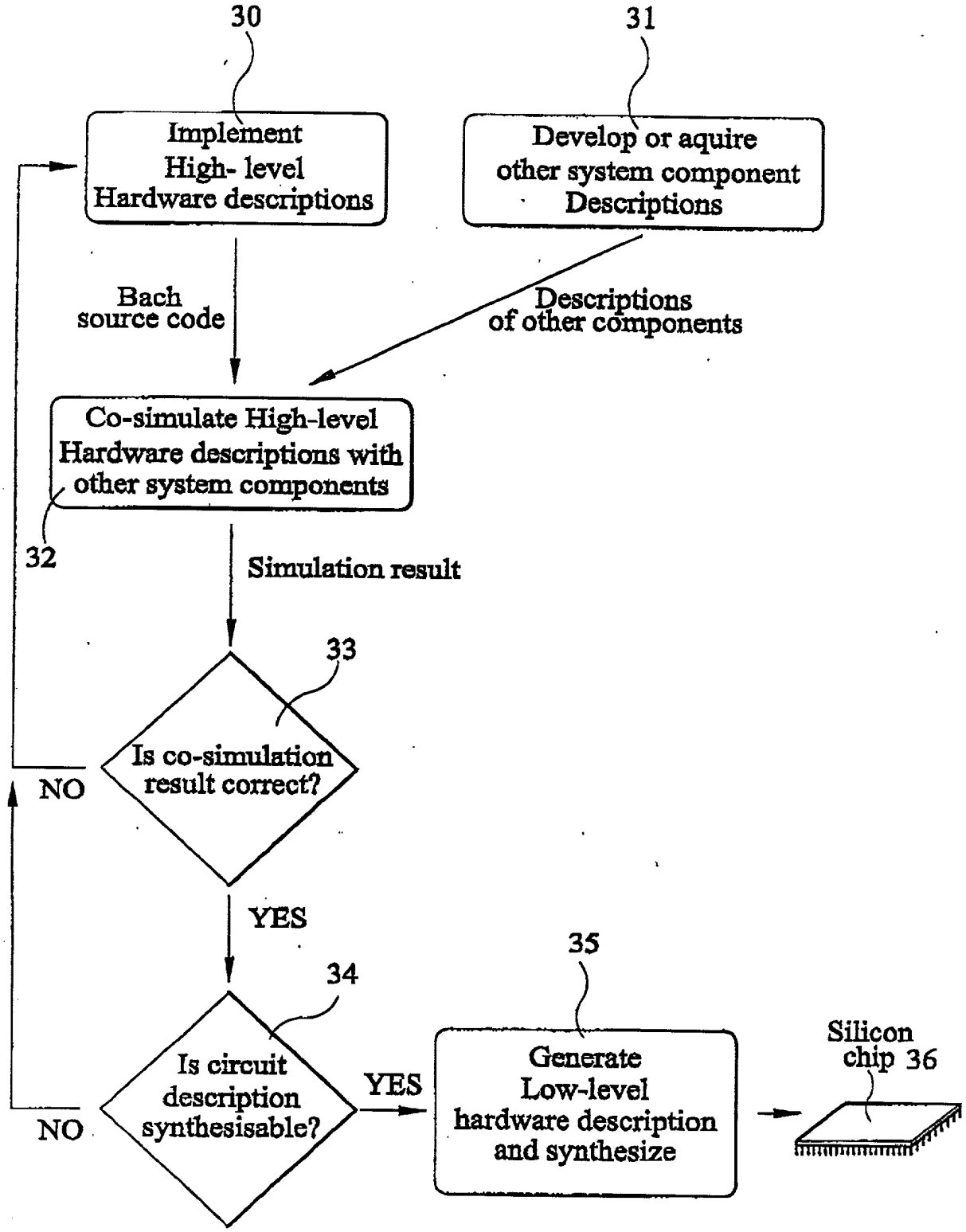

30
31
Implement
High- level
Hardware descriptions
Develop or aquire
other system component
Descriptions
Bach
source code
Descriptions
of other components
Co-simulate High-level
Hardware descriptions with
other system components
32
Simulation result
33
Is co-simulation
result correct?
NO
YES
34
35
Is circuit
description
synthesisable?
NO
YES
Generate
Low-level
hardware description
and synthesize
Silicon
chip 36

FIG.1

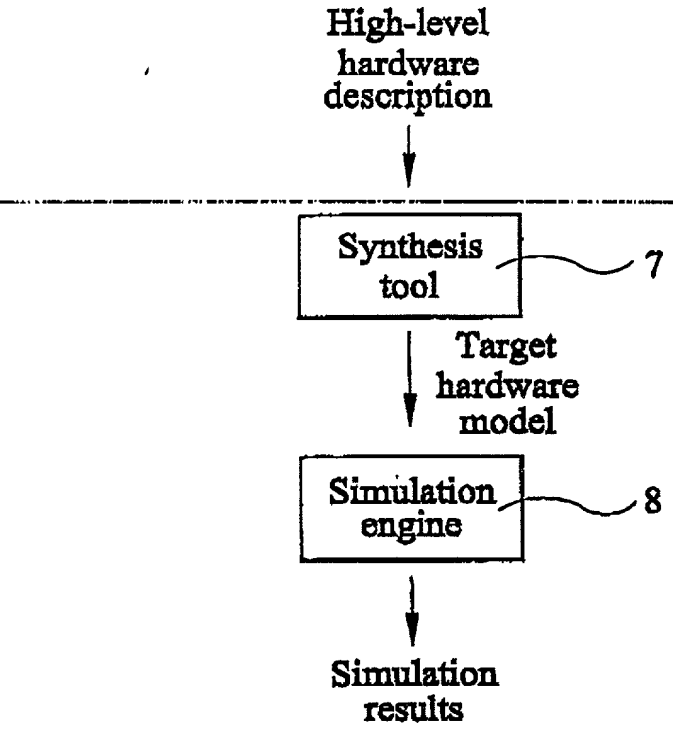

High-level hardware description
Synthesis tool
7
Target hardware model
Simulation engine
8
Simulation results

FIG.2

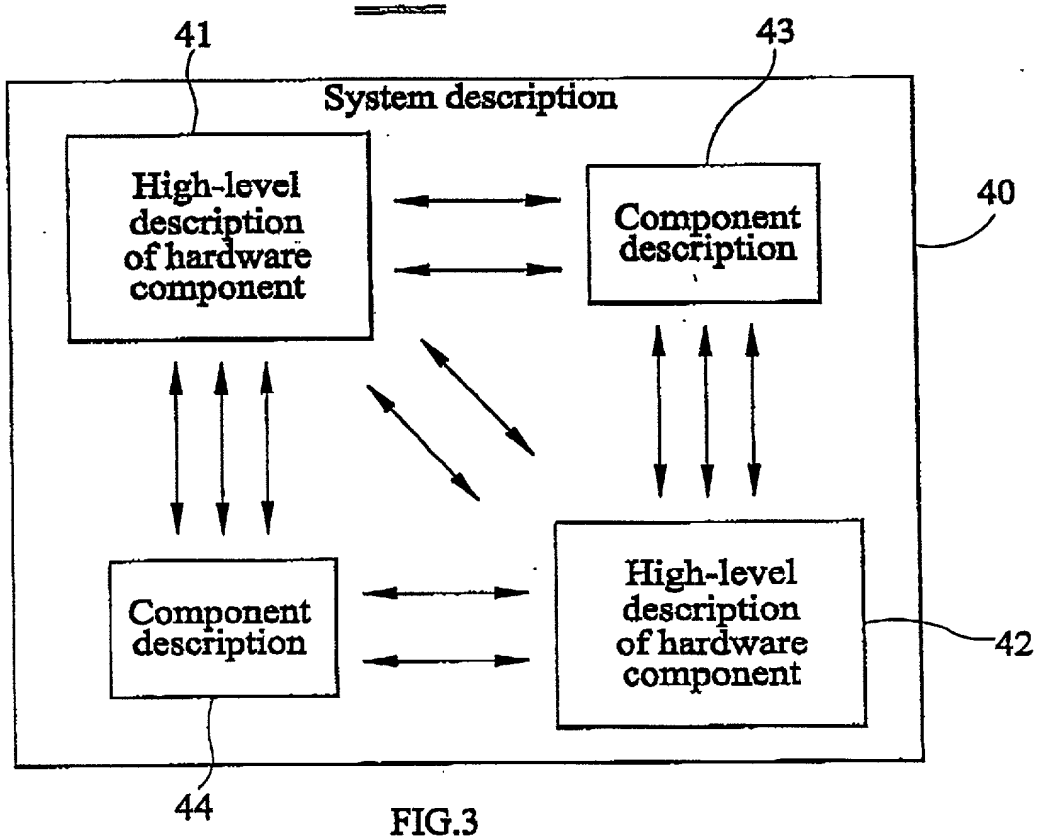

41
43
System description
High-level description of hardware component
Component description
40
Component description
High-level description of hardware component
42
44

FIG.3

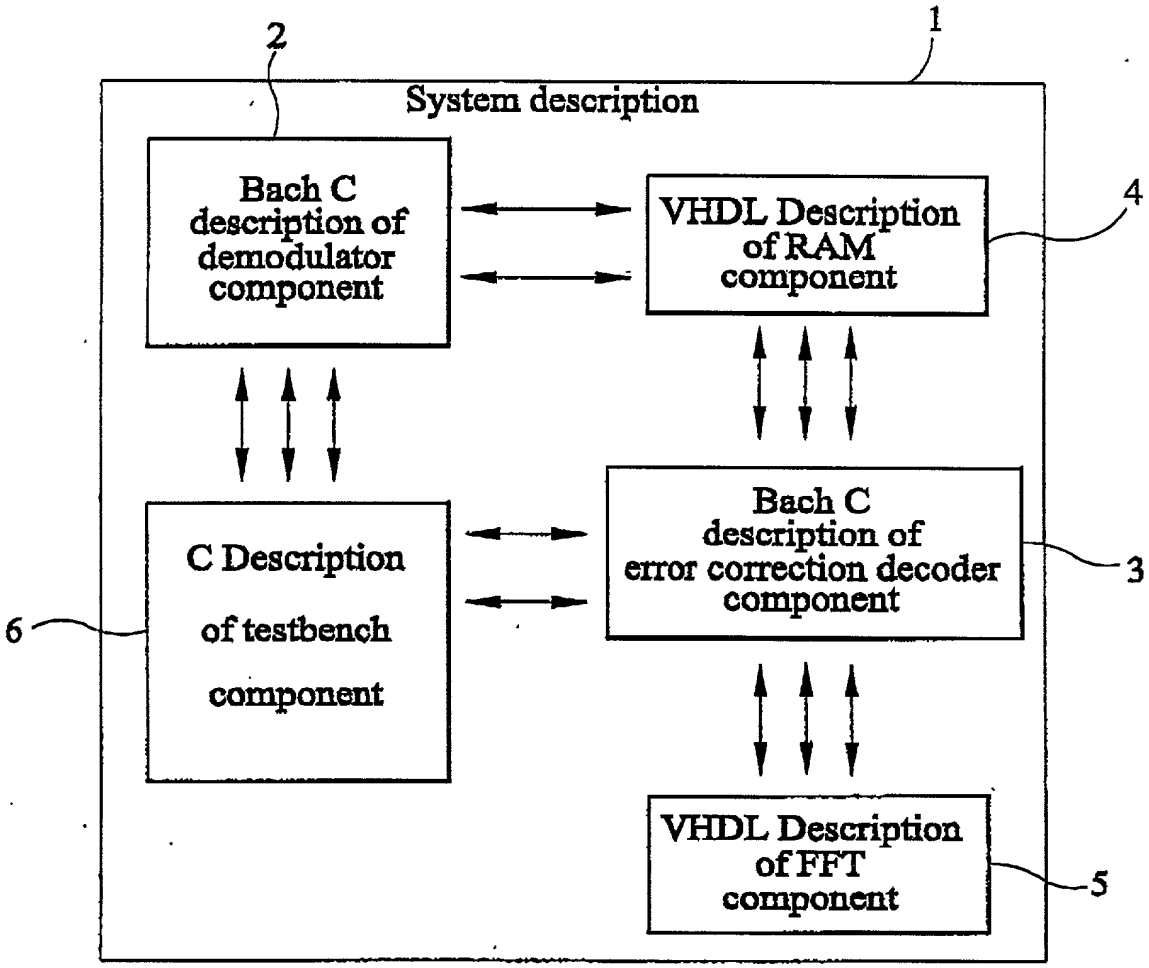

1
2
System description
Bach C description of demodulator component
VHDL Description of RAM component
4
C Description of testbench component
6
Bach C description of error correction decoder component
3
VHDL Description of FFT component
5

FIG.4

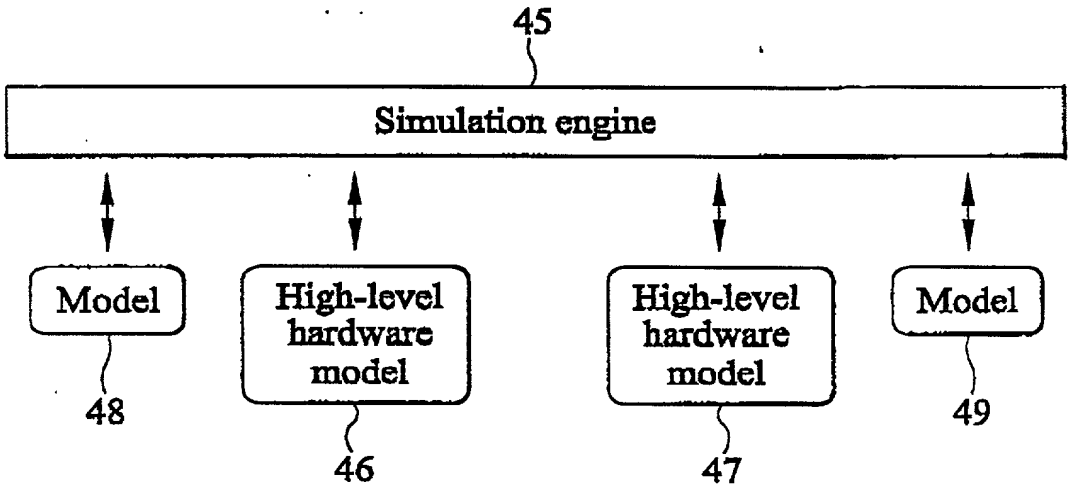

45
Simulation engine
Model
48
High-level hardware model
46
High-level hardware model
47
Model
49

FIG.5

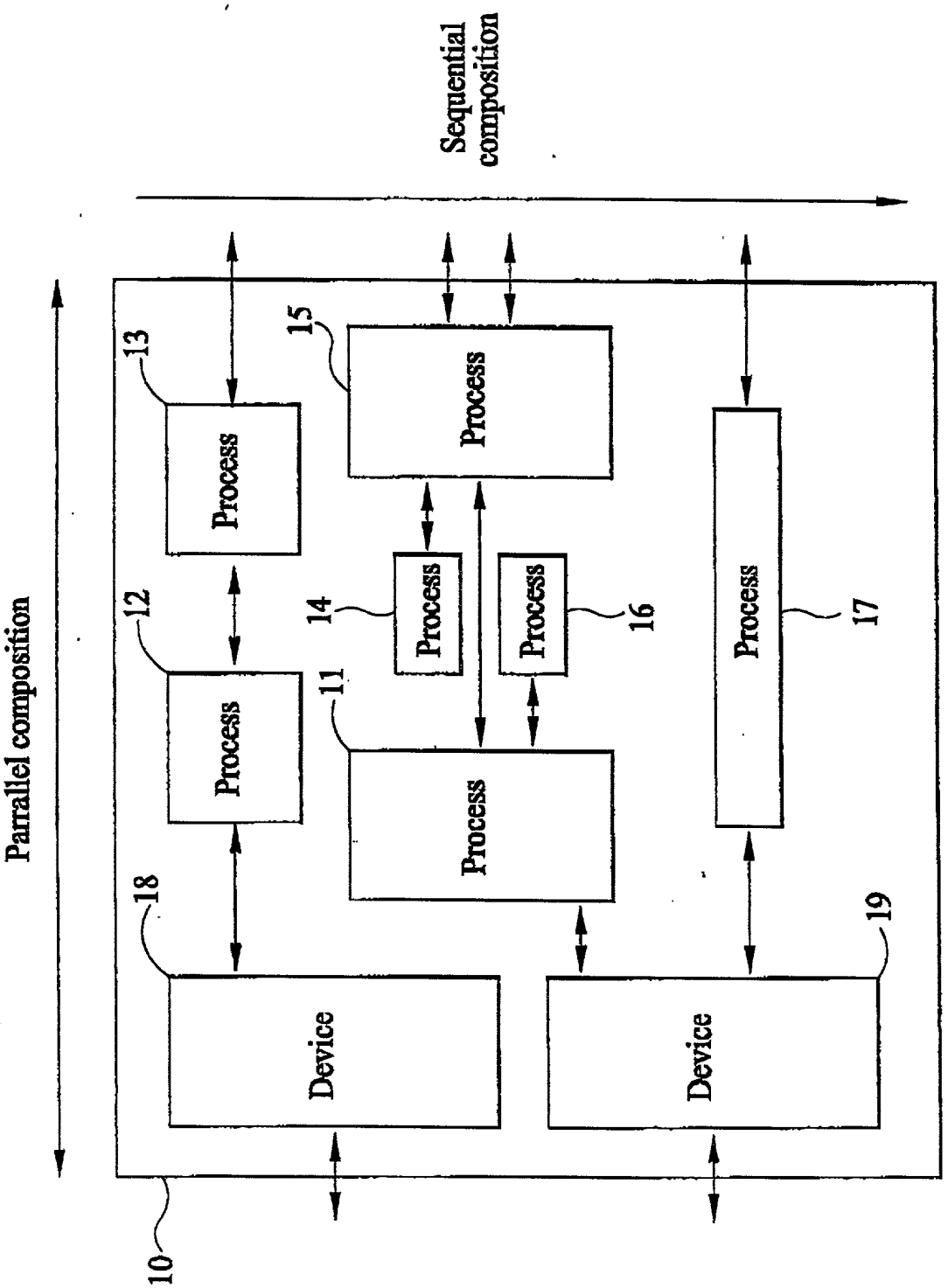

Sequential composition
Parrallel composition
Process
Process
Process
Process
Process
Process
Process
Device
Device
10
11
12
13
14
15
16
17
18
19

FIG.6

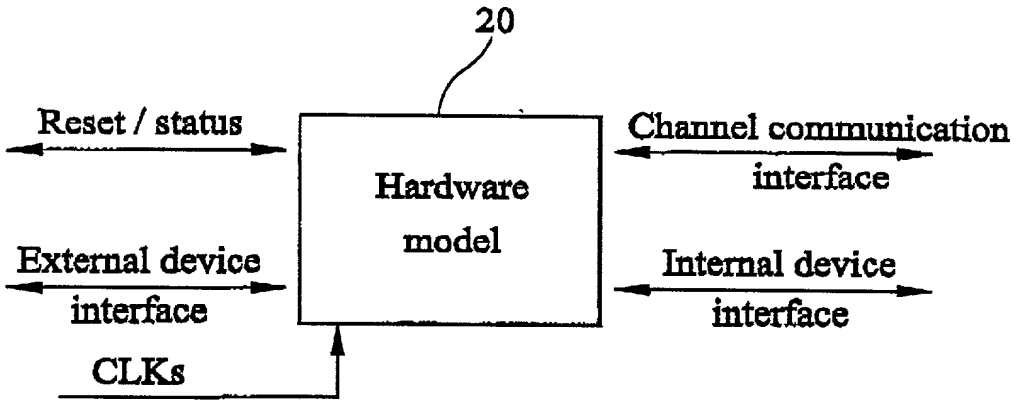

20
Reset / status
Hardware model
Channel communication interface
External device interface
Internal device interface
CLKs

FIG.7

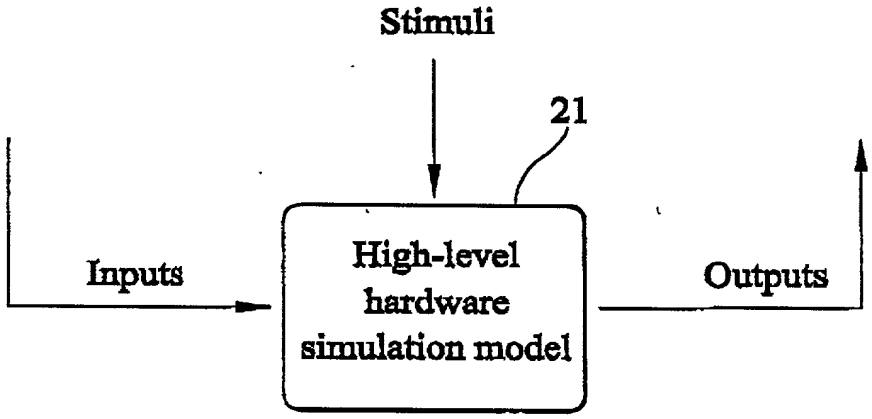

Stimuli
21
Inputs
High-level hardware simulation model
Outputs

FIG.8

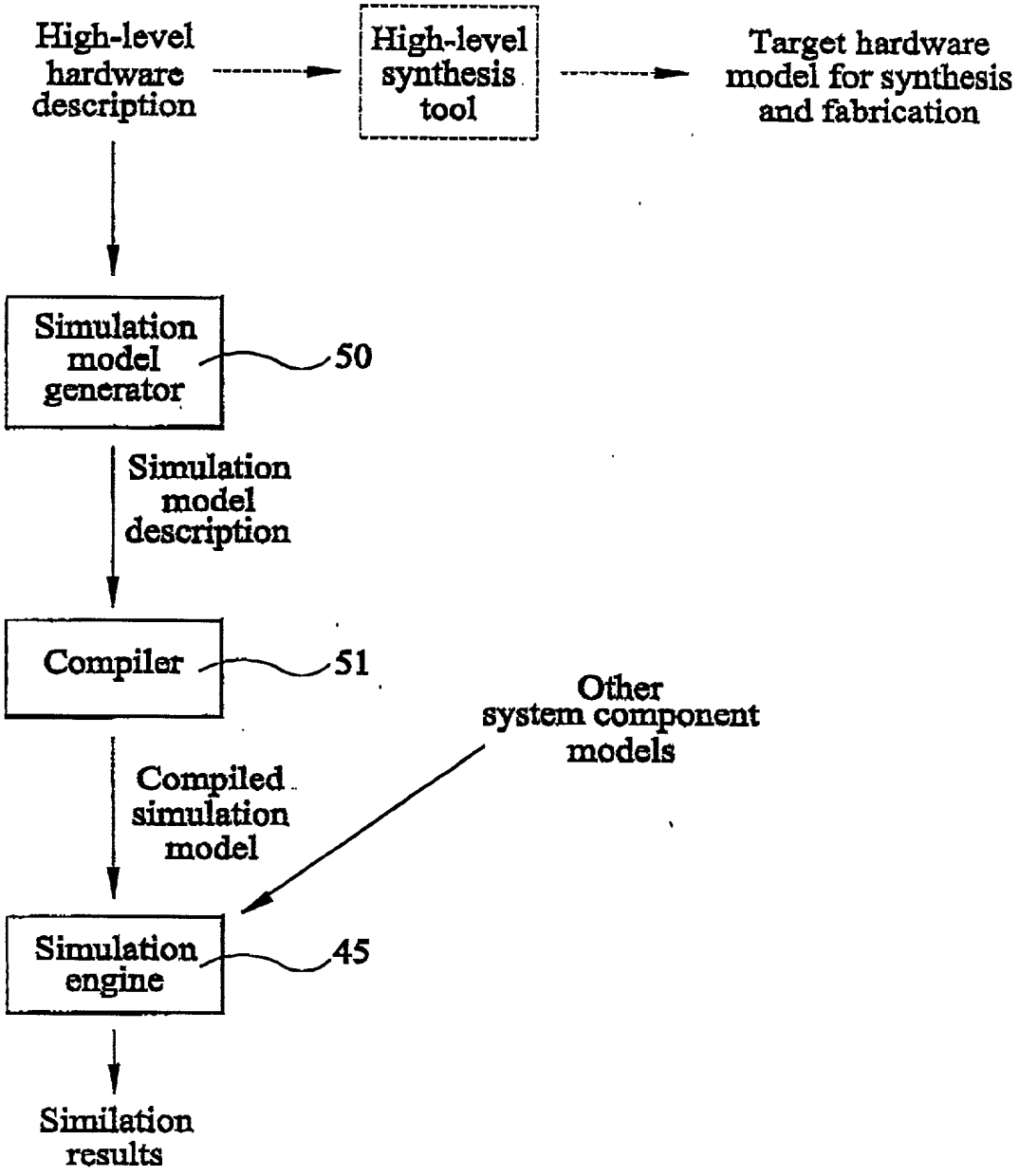

High-level hardware description
High-level synthesis tool
Target hardware model for synthesis and fabrication
Simulation model generator
50
Simulation model description
Compiler
51
Other system component models
Compiled simulation model
Simulation engine
45
Similation results

FIG.9

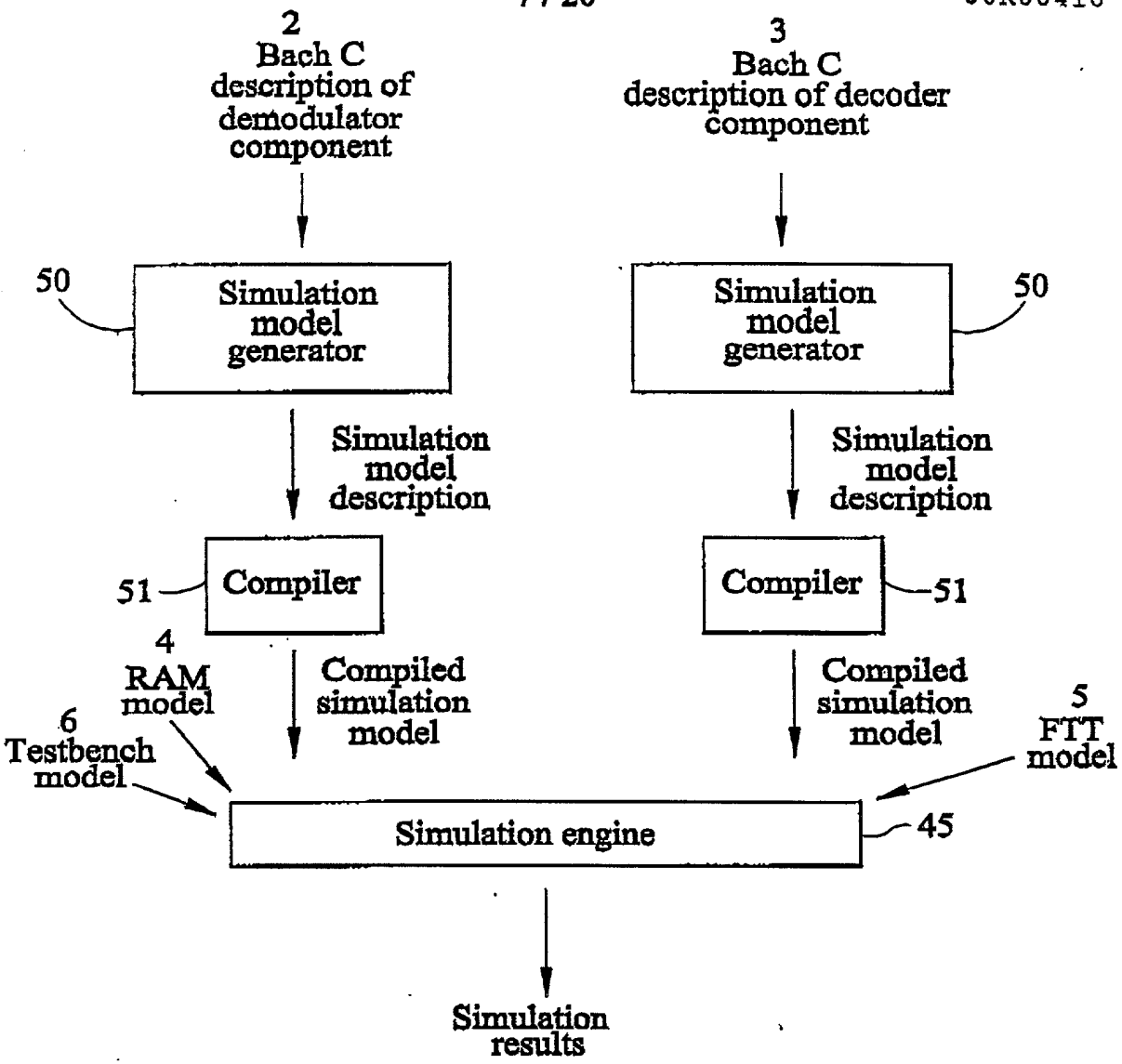

2
Bach C
description of
demodulator
component
3
Bach C
description of decoder
component
50
Simulation
model
generator
50
Simulation
model
generator
Simulation
model
description
Simulation
model
description
51
Compiler
Compiler
51
4
RAM
model
Compiled
simulation
model
Compiled
simulation
model
5
FTT
model
6
Testbench
model
Simulation engine
45
Simulation
results

FIG.10

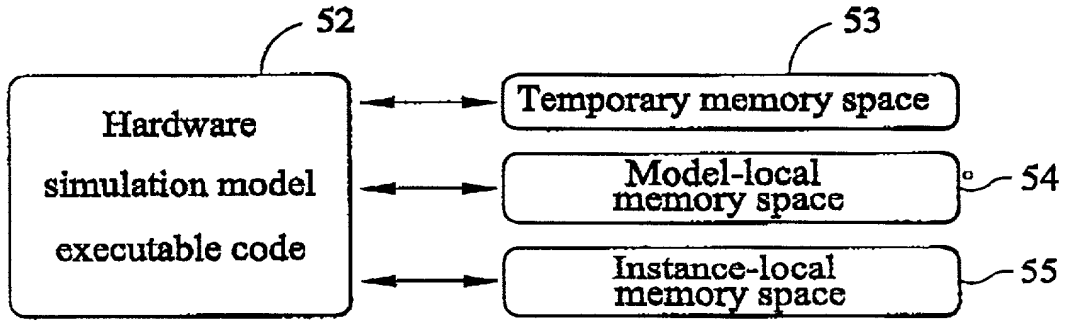

52
53
Hardware
simulation model
executable code
Temporary memory space
Model-local
memory space
54
Instance-local
memory space
55

FIG.11

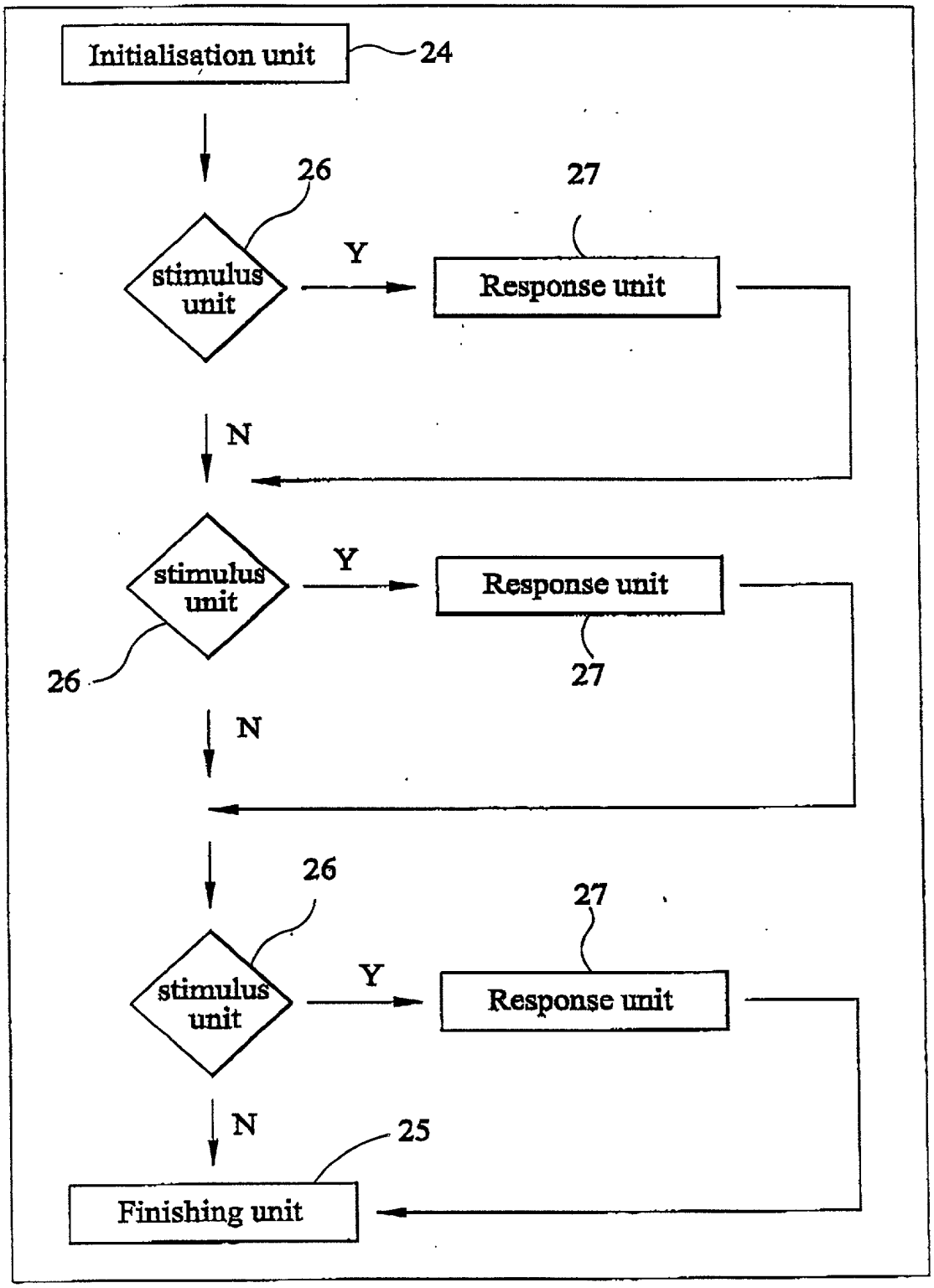

Initialisation unit
24
26
27
stimulus unit
Y
Response unit
N
stimulus unit
Y
Response unit
26
27
N
26
27
stimulus unit
Y
Response unit
N
25
Finishing unit

FIG.12

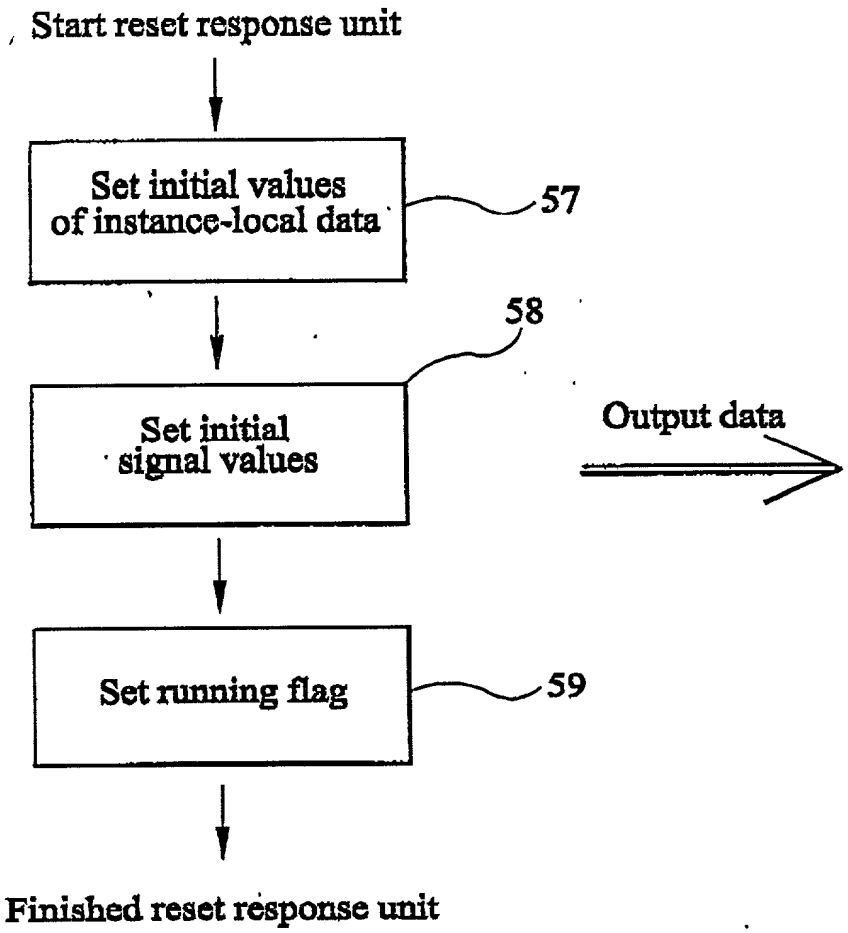

Start reset response unit
Set initial values of instance-local data
57
58
Set initial signal values
Output data
Set running flag
59
Finished reset response unit

FIG.13

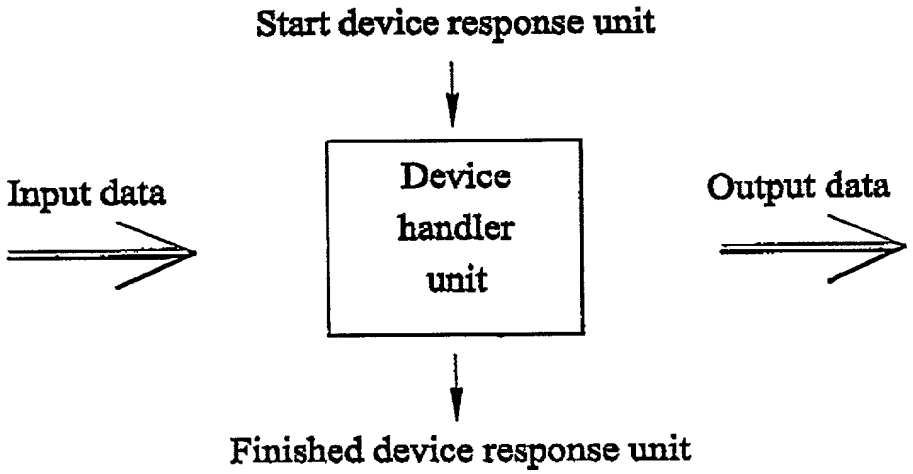

Start device response unit
Input data
Device handler unit
Output data
Finished device response unit

FIG.14

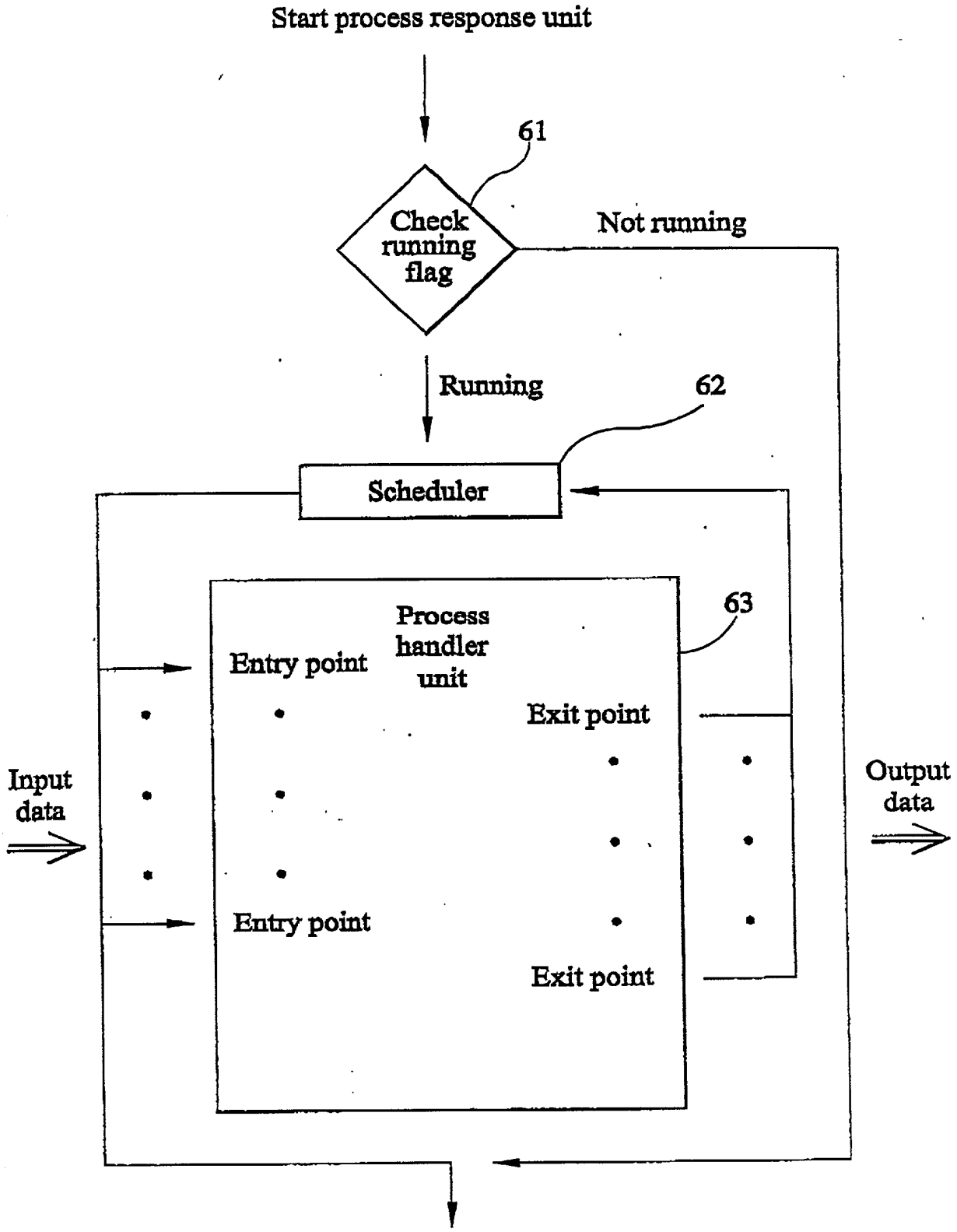

Start process response unit
61
Check running flag
Not running
Running
62
Scheduler
63
Process handler unit
Entry point
Exit point
Input data
Output data
Entry point
Exit point
Finished process response unit

FIG.15

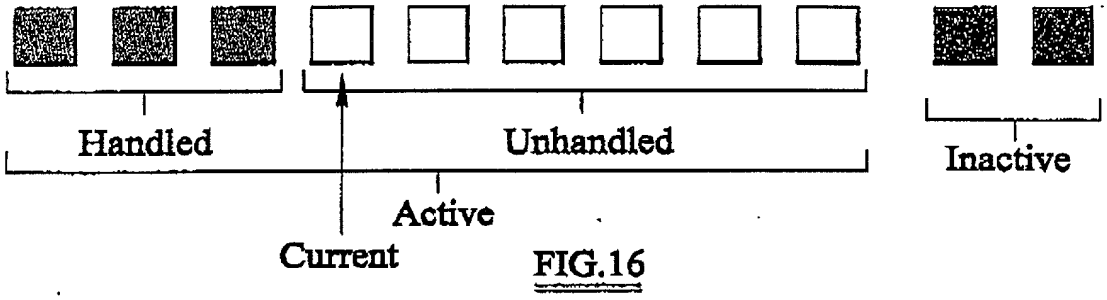

Handled
Unhandled
Inactive
Active
Current

FIG.16

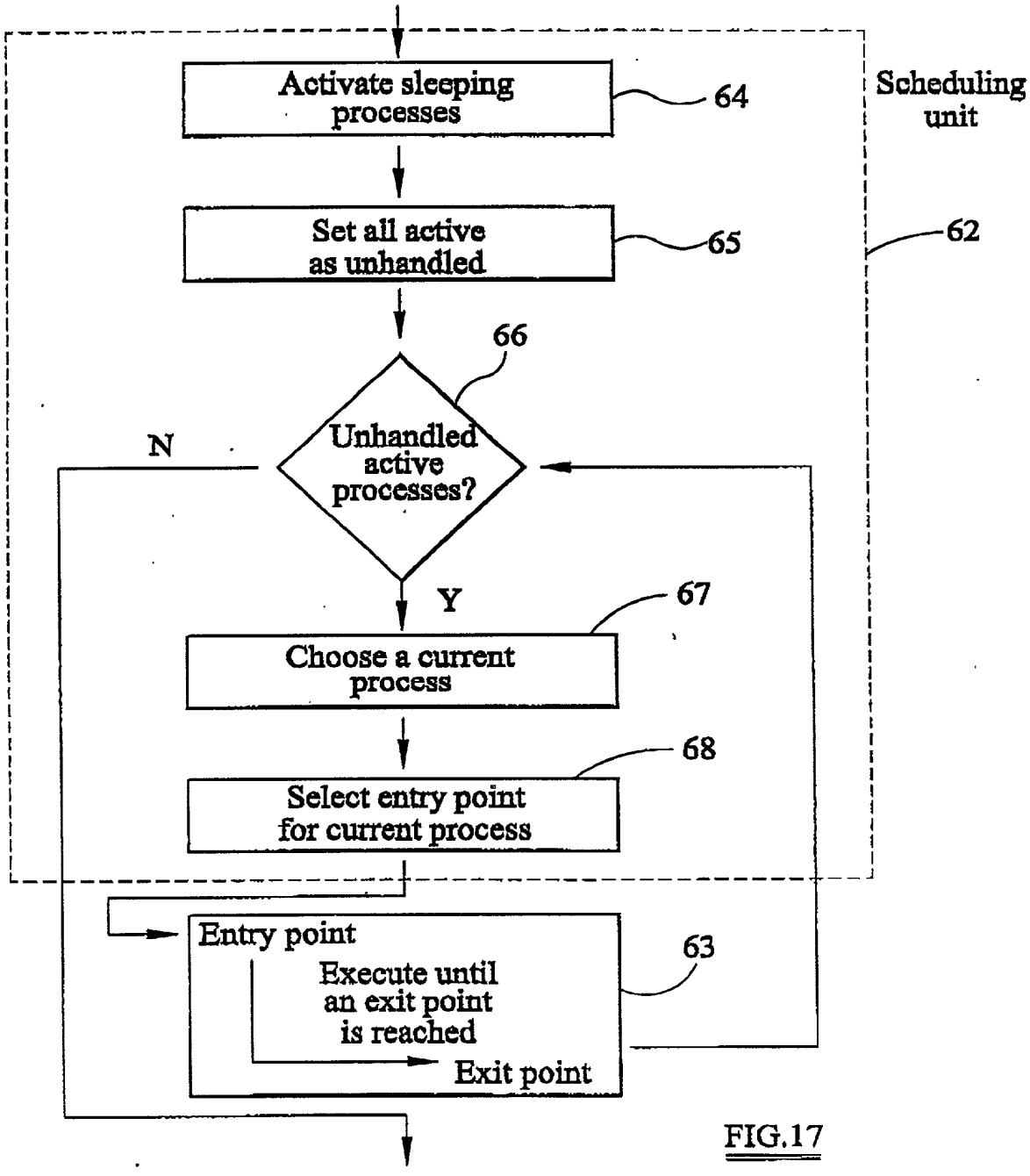

Activate sleeping processes
64
Scheduling unit
Set all active as unhandled
65
62
66
Unhandled active processes?
N
Y
67
Choose a current process
68
Select entry point for current process
Entry point
Execute until an exit point is reached
Exit point
63

FIG.17

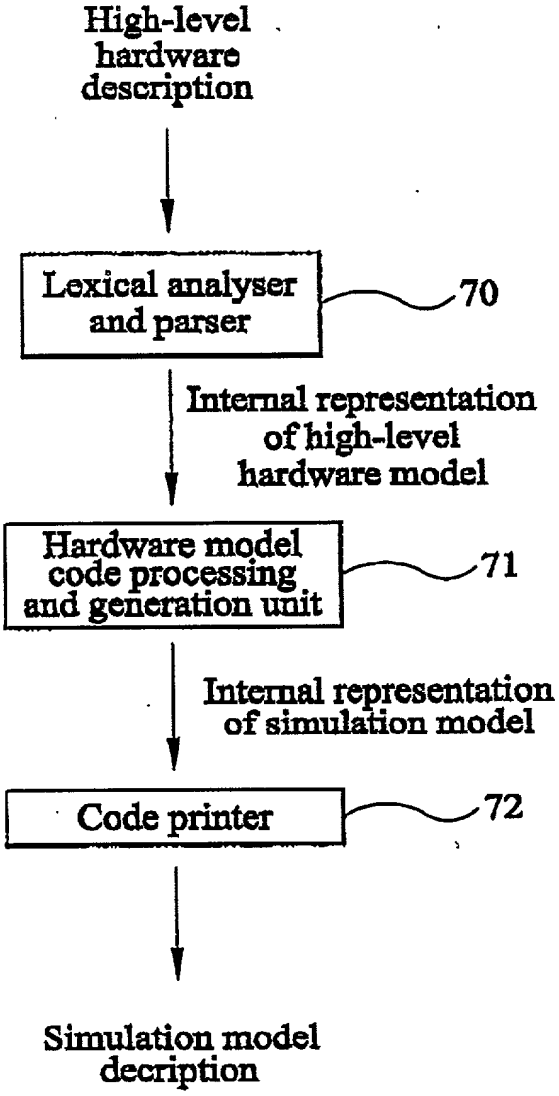

High-level hardware description
Lexical analyser and parser
70
Internal representation of high-level hardware model
Hardware model code processing and generation unit
71
Internal representation of simulation model
Code printer
72
Simulation model decription

FIG.18

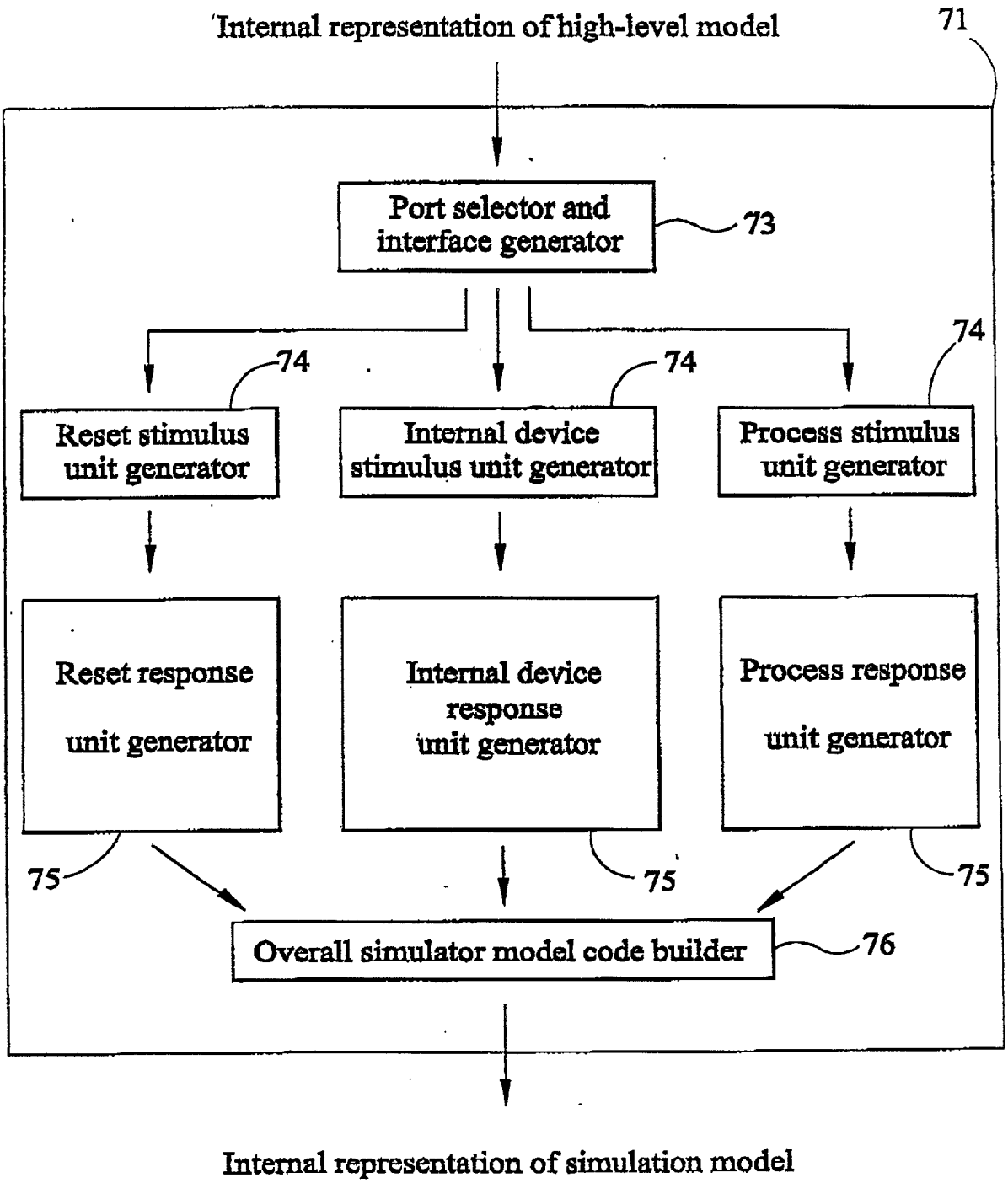

Internal representation of high-level model
71
Port selector and interface generator
73
74
74
74
Reset stimulus unit generator
Internal device stimulus unit generator
Process stimulus unit generator
Reset response unit generator
Internal device response unit generator
Process response unit generator
75
75
75
Overall simulator model code builder
76
Internal representation of simulation model

FIG.19

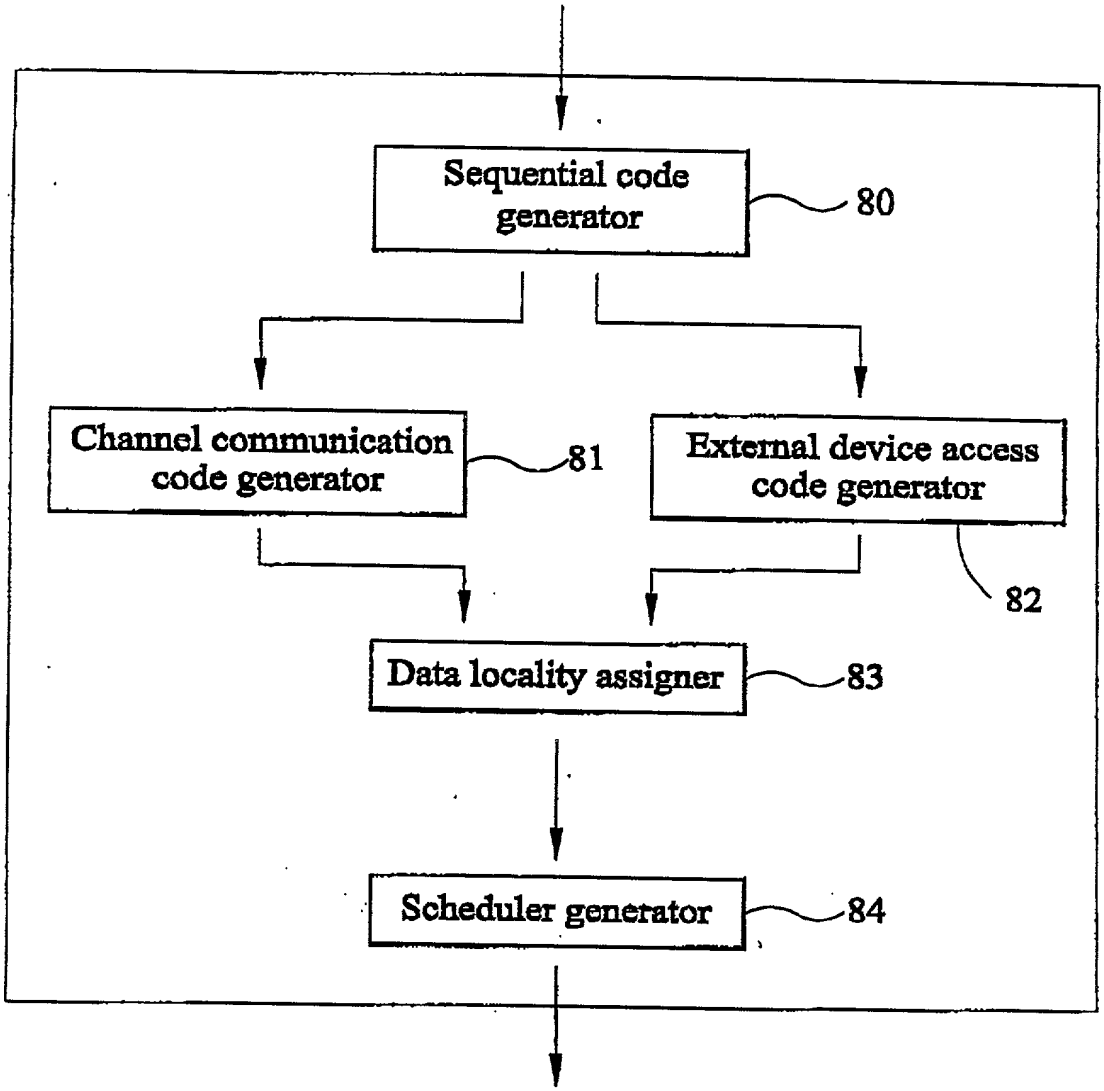

Sequential code generator
80
Channel communication code generator
81
External device access code generator
82
Data locality assigner
83
Scheduler generator
84

FIG.20

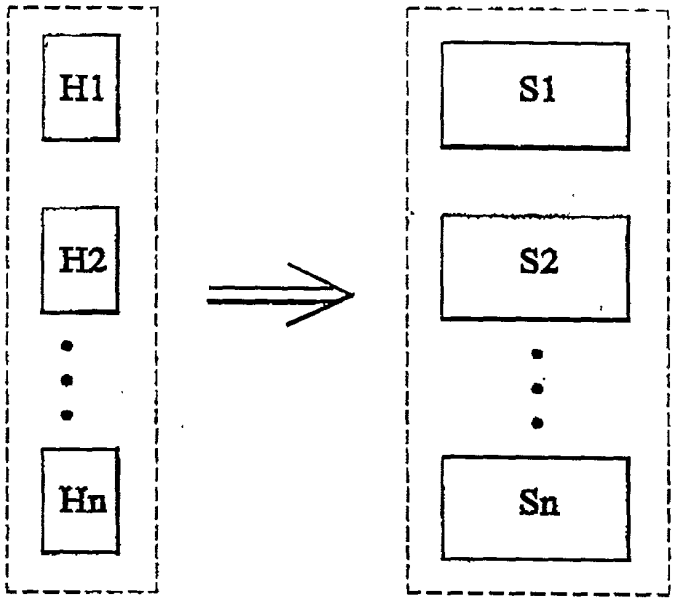


FIG.21

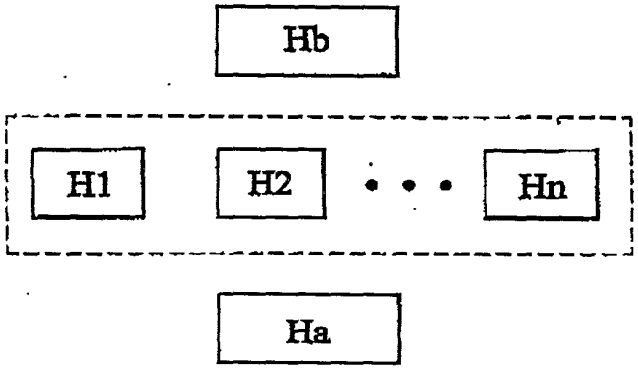


FIG.22

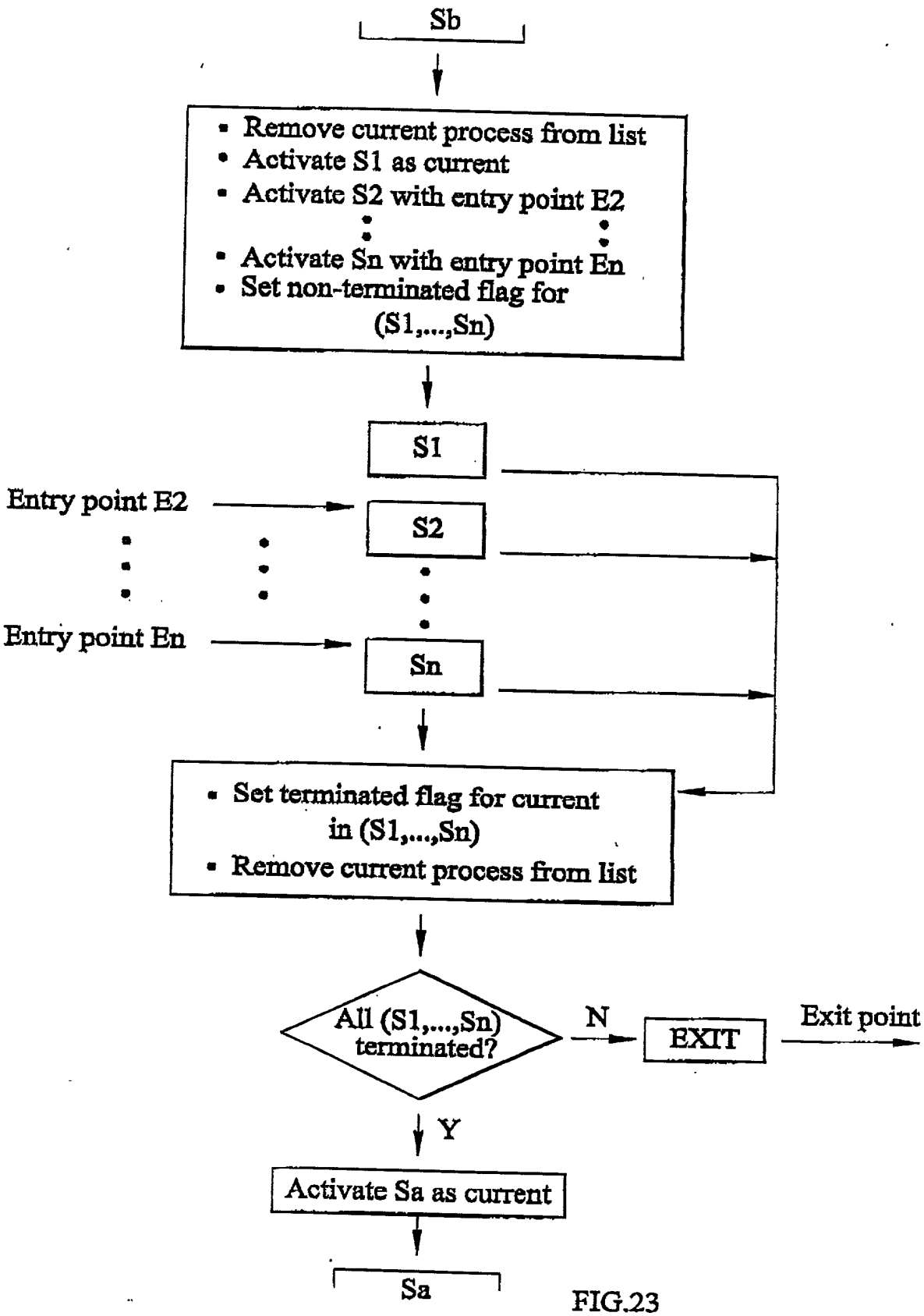

Sb
• Remove current process from list
• Activate S1 as current
• Activate S2 with entry point E2
• Activate Sn with entry point En
• Set non-terminated flag for (S1,...,Sn)
S1
Entry point E2
S2
Entry point En
Sn
• Set terminated flag for current in (S1,...,Sn)
• Remove current process from list
All (S1,...,Sn) terminated?
N
EXIT
Exit point
Y
Activate Sa as current
Sa

FIG.23

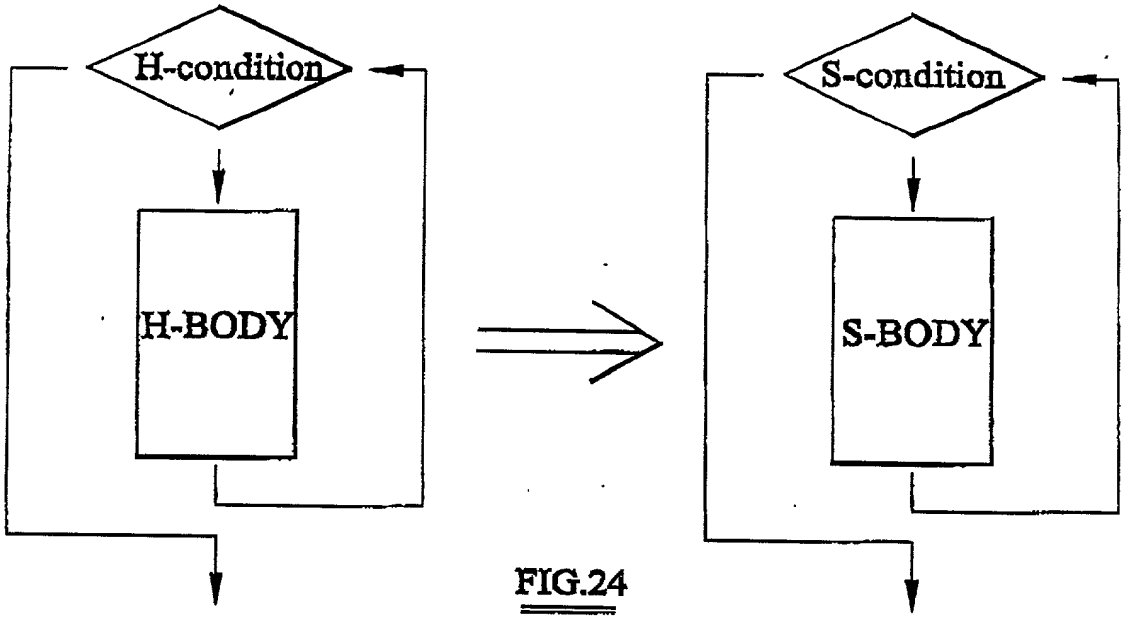

H-condition
H-BODY
S-condition
S-BODY

FIG.24

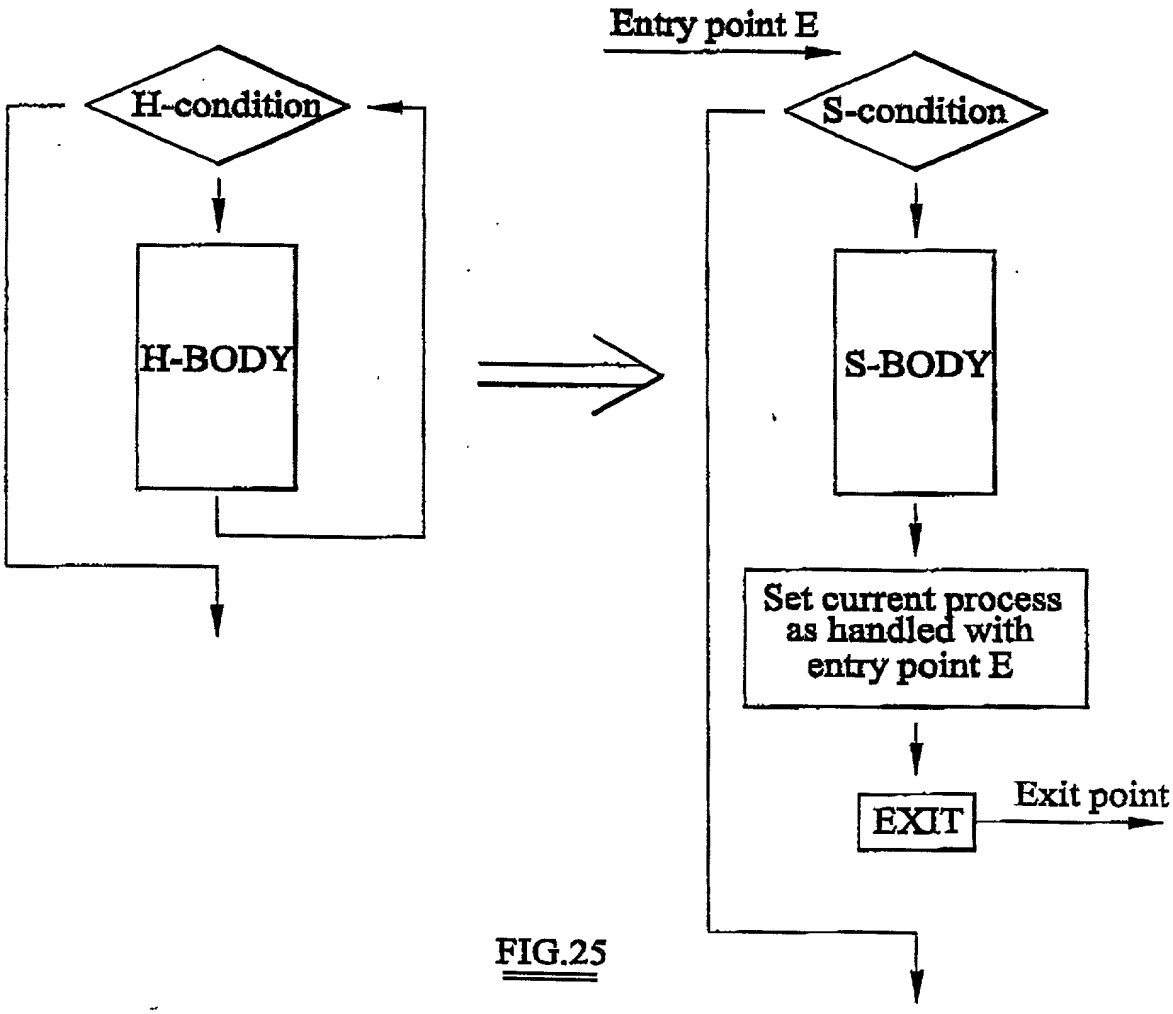

Entry point E
H-condition
H-BODY
S-condition
S-BODY
Set current process as handled with entry point E
EXIT
Exit point

FIG.25

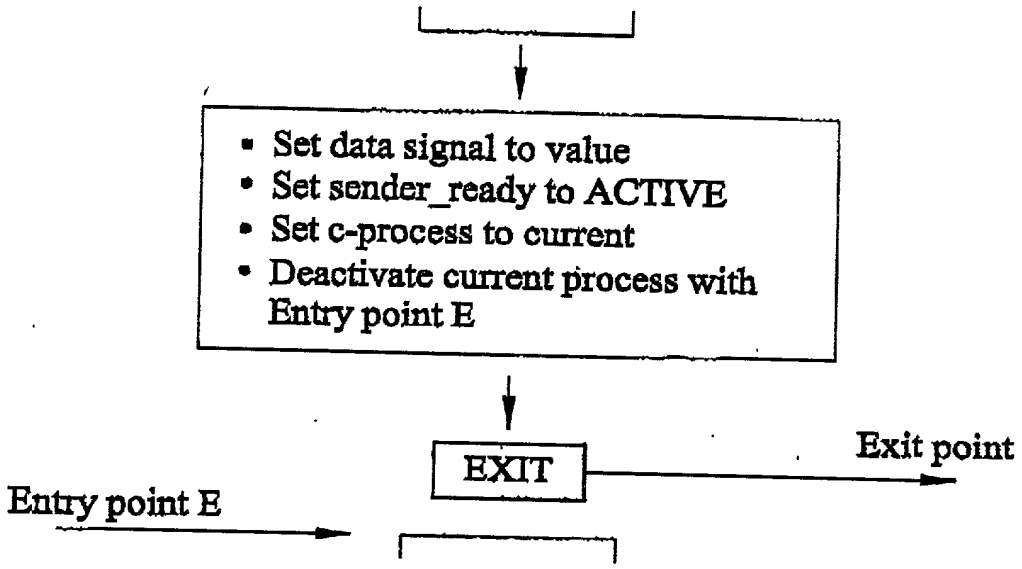

• Set data signal to value
• Set sender_ready to ACTIVE
• Set c-process to current
• Deactivate current process with Entry point E
EXIT
Exit point
Entry point E

FIG.26

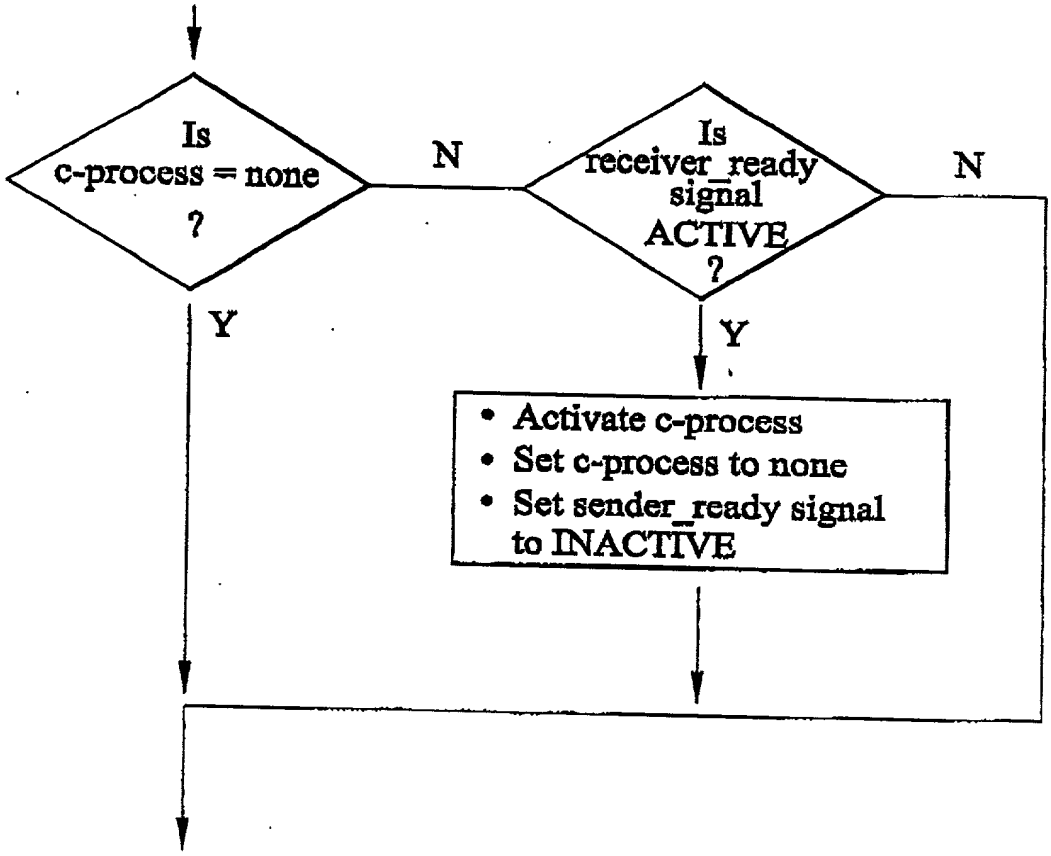

Is c-process = none ?
N
Is receiver_ready signal ACTIVE ?
N
Y
Y
• Activate c-process
• Set c-process to none
• Set sender_ready signal to INACTIVE

FIG.27

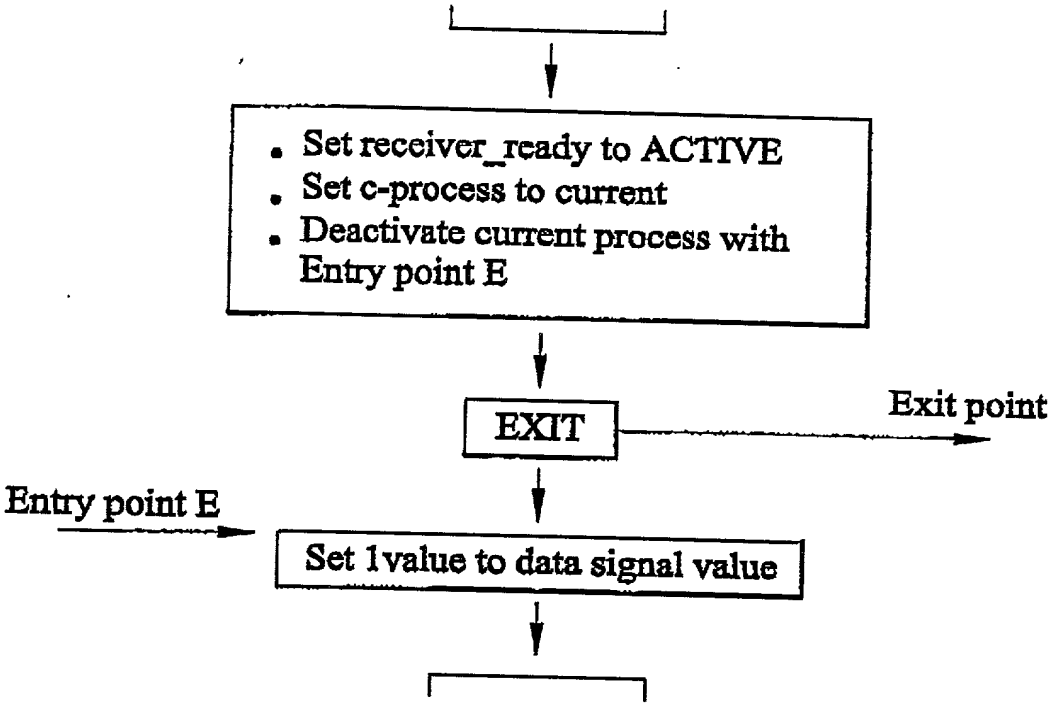

• Set receiver_ready to ACTIVE
• Set c-process to current
• Deactivate current process with Entry point E
EXIT
Exit point
Entry point E
Set 1value to data signal value

FIG.28

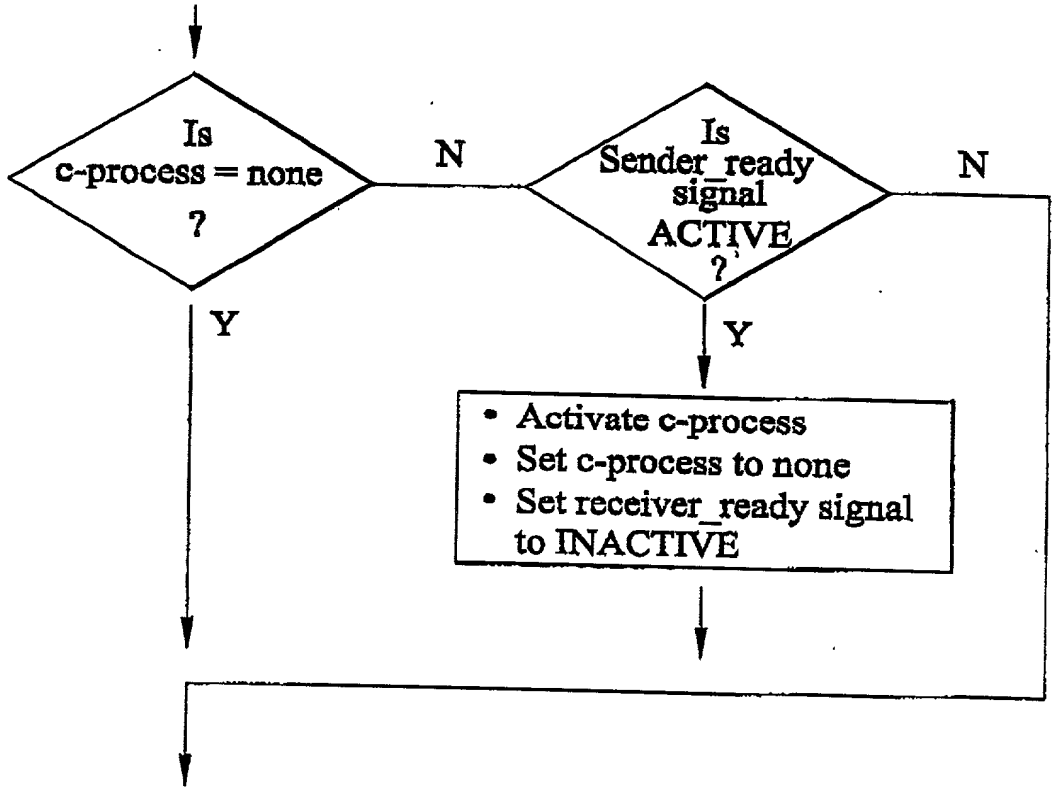
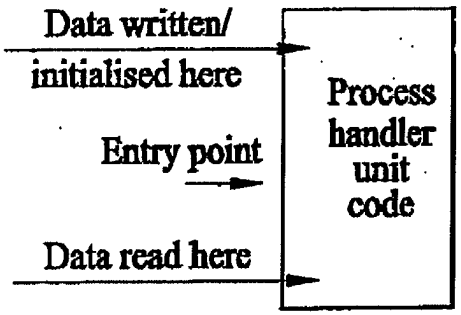
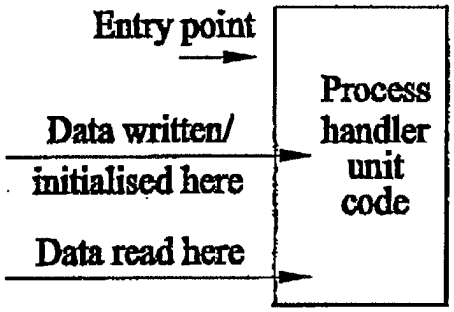

Is c-process = none ?
N
Is Sender_ready signal ACTIVE ?
N
Y
Y
• Activate c-process
• Set c-process to none
• Set receiver_ready signal to INACTIVE

FIG.29



Need for instance local data



No need for instance local data

FIG.30